



City Research Online

City, University of London Institutional Repository

Citation: Bounareli, S., Tzelepis, C., Argyriou, V., Patras, I. & Tzimiropoulos, G. (2024). HyperReenact: One-Shot Reenactment via Jointly Learning to Refine and Retarget Faces. 2023 IEEE/CVF International Conference on Computer Vision (ICCV), doi: 10.1109/ICCV51070.2023.00657 ISSN 1550-5499 doi: 10.1109/ICCV51070.2023.00657

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/31340/>

Link to published version: <https://doi.org/10.1109/ICCV51070.2023.00657>

Copyright: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

Reuse: Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

City Research Online:

<http://openaccess.city.ac.uk/>

publications@city.ac.uk

HyperReenact: One-Shot Reenactment via Jointly Learning to Refine and Retarget Faces

Stella Bounareli¹, Christos Tzelepis², Vasileios Argyriou¹, Ioannis Patras²,
Georgios Tzimiropoulos²

¹ School of Computer Science and Mathematics, Kingston University London

² School of Electronic Engineering and Computer Science, Queen Mary University of London

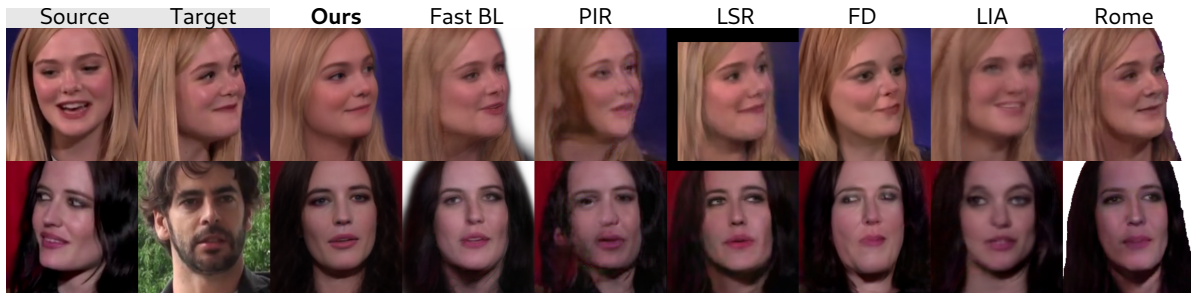


Figure 1: The proposed method, named HyperReenact, aims to synthesize realistic talking head sequences of a source identity driven by a target facial pose (i.e., 3D head orientation and facial expression). Our method performs both self and cross-subject reenactment and operates under the one-shot setting (i.e., using a single source frame). We demonstrate that the proposed framework can effectively reenact the source faces without producing significant visual artifacts, even on the challenging conditions of extreme head pose difference between the source and the target images (first row) and on cross-subject reenactment (second row). We compare our method against several state-of-the-art works on neural face reenactment, namely Fast BL [63], PIR [37], LSR [29], FD [9], LIA [54] and Rome [26].

Abstract

In this paper, we present our method for neural face reenactment, called HyperReenact, that aims to generate realistic talking head images of a source identity, driven by a target facial pose. Existing state-of-the-art face reenactment methods train controllable generative models that learn to synthesize realistic facial images, yet producing reenacted faces that are prone to significant visual artifacts, especially under the challenging condition of extreme head pose changes, or requiring expensive few-shot fine-tuning to better preserve the source identity characteristics. We propose to address these limitations by leveraging the photorealistic generation ability and the disentangled properties of a pretrained StyleGAN2 generator; by first inverting the real images into its latent space and then using a hypernetwork to perform: (i) refinement of the source identity characteristics and (ii) facial pose re-targeting, eliminating this way the dependence on external editing methods that typically produce artifacts. Our method operates under the one-shot setting (i.e., using a single source frame) and allows for cross-subject reenactment, without

requiring any subject-specific fine-tuning. We compare our method both quantitatively and qualitatively against several state-of-the-art techniques on the standard benchmarks of VoxCeleb1 and VoxCeleb2, demonstrating the superiority of our approach in producing artifact-free images, exhibiting remarkable robustness even under extreme head pose changes. We make the code and the pretrained models publicly available at: <https://github.com/StelaBou/HyperReenact>.

1. Introduction

The recent developments in deep learning and generative models [24, 25] have led to remarkable progress in the field of facial image synthesis and editing. Among the tasks that have drawn benefit from this progress is neural face reenactment, that aims to synthesize photorealistic head avatars. Specifically, given a source and a target image, the goal of face reenactment is to generate a new image that conveys the identity characteristics of the source face and the facial pose (defined as the 3D head orientation and facial expres-

sion) of the target face. The key objectives of this task are three-fold: (i) creating realistic facial images that resemble the real ones, (ii) preserving the source identity characteristics, such as the facial shape, and (iii) faithfully transferring the target facial pose. This technology is an essential component within numerous applications of augmented and virtual reality, as well as arts and entertainment industries. However, despite the recent advancements, most of the existing reenactment methods fail in producing realistic facial images in the *one-shot* setting (i.e., using a single source frame) or under *extreme head pose* movements (i.e., large differences in the head pose of the source and the target).

The majority of the state-of-the-art methods in neural face reenactment (e.g., [63, 29, 26]) train controllable models that learn to synthesize realistic images. However, these methods are prone to severe visual artifacts, especially when the source and the target faces have large head pose differences. Most of these methods rely on paired data training (i.e., images of the same identity), limiting their applicability in cross-subject reenactment. Several methods [64, 11, 29, 21] require expensive few-shot fine-tuning (i.e., using multiple different views of the source face) in order to faithfully preserve the source identity and appearance. Another line of research leverages the exceptional generation ability of pretrained generative adversarial networks (GANs) [9, 10, 62], achieving to effectively disentangle the identity from the facial pose. However, these works rely on external real image GAN inversion methods and, thus, are bounded by their limitations, such as poor identity reconstruction and image editability [47].

In this paper, we draw inspiration from recent works that combine a GAN generator with a hypernetwork [18] for real image inversion, namely HyperStyle [4] and HyperInverter [14]. These methods use a hypernetwork [18], conditioned on features derived from the original image and its initial inversion, that learns to modify the weights of the generator to obtain improved image reconstruction quality. Subsequently, one can perform semantic editing of the refined image in the latent space and produce the edited image using the updated generator weights. In spite of their high-quality reconstruction results, HyperStyle [4] and HyperInverter [14] fail upon applying global editings on the inverted images and, consequently, they are not practically applicable to neural face reenactment.

In order to address the limitations of state-of-the-art works, we tackle the neural face reenactment task by leveraging the photorealistic image generation and the disentangled properties of a pretrained StyleGAN2 [25], along with a hypernetwork [18]. We present a novel method that performs both faithful identity reconstruction and effective facial image editing by learning to update the weights of a StyleGAN2 generator using a hypernetwork approach. Specifically, our model effectively combines the appearance

features of a source image and the facial pose features of a target image to create new facial images that preserve the source identity and convey the target facial pose.

Overall, the main contributions of this paper can be summarized as follows:

1. We present a novel framework for face reenactment that leverages the adaptive nature of hypernetworks [18] to alter the weights of a powerful StyleGAN2 [25], so as to perform both: (i) refinement of the source identity details and (ii) reenactment of the source face in the target facial pose. To the best of our knowledge, we are the first to show the effectiveness of merging the steps of inversion refinement and facial pose editing for robust and realistic face reenactment.
2. We demonstrate that our method is able to successfully operate under one-shot settings (i.e. using *a single* source frame) without requiring any fine-tuning, preserving the source identity characteristics on both self and cross-subject reenactment scenarios. This holds true even in challenging cases where the source face is partially self-occluded (i.e., in partial facial views due to highly non-frontal head poses).
3. We show that our method achieves state-of-the-art results even on *extreme* head pose variations, generating artifact-free images and exhibiting remarkable robustness to large head pose shifts.
4. We conduct experiments on the standard benchmarks of VoxCeleb1 and VoxCeleb2 [30, 12], performing qualitative and quantitative comparisons with existing state-of-the-art reenactment techniques. We show that our proposed method achieves compelling results both on identity preservation and facial pose transfer.

2. Related work

Facial image editing Several recent methods [42, 46, 49, 6, 31, 32, 59, 2, 35] leverage the remarkable ability of modern pretrained GAN models (e.g., StyleGAN2 [25]) in producing photorealistic facial images in order to edit various facial attributes, such as head pose, facial expressions or hair style. The key idea of such methods lies in learning to manipulate the latent representations (in \mathcal{W} , \mathcal{W}^+ or \mathcal{S} latent space) of a StyleGAN2 generator in order to generate meaningful editings of different semantics on the synthetic images. The existing unsupervised methods [51, 49] are able to find disentangled directions that edit different semantics on the facial images, albeit without providing any controllability on the manipulation of the images. In order to allow for explicit control over the image editing, several approaches [2, 10, 46, 35, 48, 61] rely on external supervision from pretrained models, such as 3D Morphable Models

(3DMM) [8], or vision-language models [36]. Despite their effectiveness in editing synthetic images, such methods fail to manipulate effectively real images – i.e., having to perform editing in the non-native latent code provided by an external GAN inversion method, as discussed below.

Real image inversion: GAN inversion methods [57, 47, 45] allow for encoding of real images into the latent space of pretrained generators, which is required at the same time to allow for semantic image editing. The main challenges of real image inversion are to (i) faithfully reconstruct the real images and (ii) enable facial image editing without producing visual artifacts. This is typically referred to as the “reconstruction-editability” trade-off. The existing inversion methods mainly focus either on optimization-based approaches, which require expensive iterative optimization for each image, rendering them not-applicable for real-time applications, or on encoder-based architectures.

Encoder-based methods [3, 38, 47, 52, 5] train encoders that learn to predict the latent code that best reconstructs the real image. While providing better semantic editing than optimization-based approaches, encoder-based methods fail in faithfully reconstructing real images (i.e., by missing crucial identity details). In order to coordinate the trade-off between reconstruction quality, editability, and inference time, some recent works [4, 14] propose to optimize a hypernetwork [18] that learns to update the weights of a pretrained GAN generator so as to refine any missing identity details. HyperStyle [4] first inverts the real images into the latent space of StyleGAN2 (\mathcal{W} or \mathcal{W}^+) using a pretrained encoder-based inversion method [47] and then trains a hypernetwork that, given a pair of a real and an initial reconstructed image, predicts an offset Δ_ℓ for each layer of the generator. Both HyperStyle [4] and HyperInverter [14] lead in high quality reconstructions, albeit suffering from many visual artifacts in the case of head pose editing.

Neural face reenactment A recent line of works focus on learning disentangled representations for the identity and the facial pose using facial landmarks [64, 63, 22, 19, 65]. However, such methods perform poorly on the challenging task of cross-subject reenactment, since facial landmarks preserve the facial shape and consequently the identity geometry of the target face. In order to mitigate the identity leakage from the target face to the source face, several methods [28, 17, 46, 9, 15, 37, 26, 60] leverage the disentangled properties of 3D Morphable Models (3DMM) [8, 16]. Xu et al. [58] propose a unified architecture that learns to perform both face reenactment and swapping. Warping based methods [55, 43, 53, 37, 15, 67, 62, 54] learn a motion field between the source and target frames in order to synthesize the reenacted faces. [15, 37] propose a two-stage architecture that first generates a warped image using the learned motion field and then refines the warped image to minimize the visual artifacts caused by the warping operation.

Despite their realistic results in small pose variations, such methods fail in the more challenging and realistic condition of large head pose variations (i.e., under large differences between the target and the source head pose).

A more recent line of works [9, 62, 10] propose the incorporation of the powerful pretrained StyleGAN2 model. StyleHEAT [62] proposes to control the spatial features of the pretrained StyleGAN2 generator using a learned motion field between the source and target frames. However, their method is trained on the HDTF dataset [67], i.e., a video dataset with mostly frontal talking head videos, leading to poor reenactment performance on more realistic datasets, such as VoxCeleb [30, 12], which comprises of a larger distribution on the existing head poses. Bounareli et al. [9] propose to fine-tune a StyleGAN2 model (pretrained on FFHQ [24]) on VoxCeleb1 dataset [30] and then learn the linear directions that are responsible for controlling the changes in the facial pose. For editing the real images, [9] relies on the encoder-based inversion method of [47], which results in visual artifacts on large head pose variations and requires an additional optimization step [39] to refine the missing identity details. Finally, [10] proposes to disentangle the identity characteristics from the facial pose by leveraging the disentangled properties of StyleGAN2’s style space.

In this work, we also use the StyleGAN2 generator, pretrained on VoxCeleb [9], which allows for better generalization on other existing video datasets [67, 40]. To the best of our knowledge, we are the first to propose the optimization of a hypernetwork [18] based reenactment module, merging this way the steps of inversion refinement and facial pose editing towards robust and realistic face reenactment.

3. Proposed method

In this section we present the proposed HyperReenact framework for neural face reenactment. An overview of the method is shown in Fig. 2. In a nutshell, the hypernetwork \mathcal{H} , guided by the source appearance (f_{app}) and target facial pose (f_{p}) related features, learns to predict the offsets $\Delta\theta_\ell, \ell = 1, \dots, N$, where N is the total number of the layers in the generator \mathcal{G} . Then, given the initial latent code \mathbf{w}_s and the updated weights $\hat{\theta} = \theta \cdot (1 + \Delta\theta)$, the generator \mathcal{G} is able to generate an image that has the identity of the source and the facial pose of the target face.

3.1. GAN Inversion

The generator of StyleGAN2 [25] takes as input random latent codes $\mathbf{z} \in \mathbb{R}^{512}$ sampled from the standard Gaussian distribution, which are then fed to the mapping network to get the intermediate latent codes $\mathbf{w} \in \mathbb{R}^{512}$ (i.e., codes in the \mathcal{W} space). Most inversion methods (e.g., [1, 47]) invert to \mathcal{W} space or its extended $\mathcal{W}^+ \subseteq \mathbb{R}^{N \times 512}$ space, where a different latent code is fed to each of the N layer

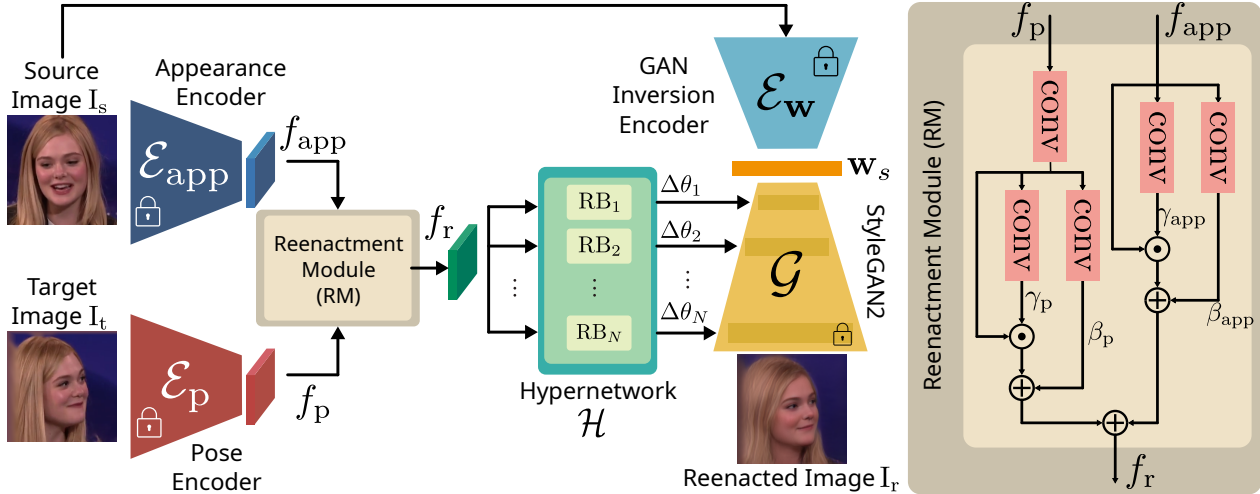


Figure 2: **HyperReenact network architecture** Given a source (I_s) and a target (I_t) image, we first extract the source appearance features, f_{app} , and the target pose features, f_p , using the appearance (\mathcal{E}_{app}) and the pose (\mathcal{E}_p) encoders, respectively. The Reenactment Module (RM) learns to effectively fuse these features, producing a feature map f_r that serves as input into each Reenactment Block (RB) of our hypernetwork module \mathcal{H} . The predicted offsets, $\Delta\theta$, update the weights of the StyleGAN2 generator \mathcal{G} so that using the inverted latent code w_s generates a new image I_r that conveys the identity characteristics of the source face and the facial pose of the target face. We note that, during training, the encoders \mathcal{E}_{app} , and \mathcal{E}_p , along with the generator \mathcal{G} are kept frozen, and we optimize only the Reenactment Module (RM) and the hypernetwork module \mathcal{H} .

of the generator. Tov et al. [47] show that, using the \mathcal{W}^+ latent space for real image inversion, results in better reconstruction quality. However, it can be challenging to edit the real images using \mathcal{W}^+ , in particular when altering the head pose, as this can lead to severe visual artifacts (e.g., [9]). In this work, in order to get an initial inverted latent code, we use e4e [47], which inverts the real images into the \mathcal{W}^+ space. We note that we use an off-the-shelf inversion model, trained on the VoxCeleb1 [30] dataset and provided by the authors of [9], for the initial inversion step, while during training the inversion encoder \mathcal{E}_w is not updated (see Fig. 2).

3.2. HyperReenact Architecture

The proposed HyperReenact aims to modify the weights θ of the generator \mathcal{G} to both lead to better reconstruction (without any further optimization steps) and perform neural face reenactment eliminating any visual artifacts on the generated images (Fig. 2). Given a pair of source and target images our framework first extracts the corresponding appearance f_{app} and facial pose f_p features. Specifically, to encode the appearance of a face we use the ArcFace [13] encoder E_{app} . ArcFace is trained on the face recognition task, as a result its features only capture the identity characteristics of a face. We note that we extract the feature map f_{app} of shape $512 \times 7 \times 7$ picking the output of the last convolutional layer of ArcFace. Similarly, to encode the facial pose of a face we use the encoder \mathcal{E}_p , which is a pre-trained 3D shape model (DECA [16]). DECA is trained on

3D facial shape model reconstruction, hence the extracted features only capture the facial pose, without taking into consideration the appearance. We extract the feature map f_p of shape $2048 \times 7 \times 7$, picking the output of the last convolutional layer of \mathcal{E}_p .

Our goal is to guide the hypernetwork with a feature map that combines the appearance features from the source image and the facial pose features from the target image. Inspired by the Spatially-Adaptive Denormalization (SPADE) module [33], we propose to *blend* the two feature maps, f_{app} and f_p , using the Reenactment Module (RM). As shown in Fig. 2, the RM includes a 1×1 convolution to project f_p into the same channel size as f_{app} , obtaining f'_p . Then, for each feature map, we learn two modulation parameters, namely γ and β . The final combined feature map f_r with size $512 \times 7 \times 7$ is calculated as:

$$f_r = \gamma_{app} \odot f_{app} + \beta_{app} + \gamma_p \odot f'_p + \beta_p, \quad (1)$$

where \odot denotes the element-wise multiplication.

Our hypernetwork module follows a similar architecture with the one in HyperStyle [4]. Specifically, it consists of $M \subset N$ Reenactment Blocks (RB), where M is the number of layers we control and N is the total number of layers of the generator. Each RB takes as input the combined feature map f_r and outputs an offset $\Delta\theta_\ell$ of size $C_\ell^{out} \times C_\ell^{in} \times 1 \times 1$. We then spatially repeat each offset by the kernel dimension of each layer $k_\ell \times k_\ell$, significantly reducing the number of learnable parameters [4]. Finally, the updated weights for

each layer ℓ of the generator are computed as $\hat{\theta}_\ell = \theta_\ell \cdot (1 + \Delta\theta_\ell)$. We provide a detailed analysis on the architecture of our hypernetwork module in the supplementary material.

3.3. Training Process

We train the proposed HyperReenact framework following a curriculum learning (CL) scheme [7], where we gradually increase the complexity of the training data. Specifically, we first train our network on real image inversion, where the source and target faces are the same. We further train our model on the task of self reenactment, where the source and target faces have the same identity but different facial pose. Finally, we continue training our model on cross-subject reenactment, where the source and target faces have different identity and facial pose. In Section 4.2, we show that the proposed curriculum learning scheme improves our results. We detail each training phase below.

Phase 1: Real image inversion On the first training phase, the source (I_s) and the target (I_t) images are the same to the input image (I). Given the appearance and facial pose of the input image I as well as the initial inverted latent code w , we train our network to refine the missing identity details between the real image and its initial reconstruction using w . Our training objective during the inversion phase consists of the following reconstruction loss terms:

$$\mathcal{L} = \lambda_{pix}\mathcal{L}_{pix} + \lambda_{lpips}\mathcal{L}_{lpips} + \lambda_{id}\mathcal{L}_{id} + \lambda_g\mathcal{L}_g, \quad (2)$$

where \mathcal{L}_{pix} is the ℓ_1 pixel-wise loss and \mathcal{L}_{lpips} is the perceptual loss [23] between the real I and the refined \hat{I} image. Additionally, to further enhance the identity preservation we calculate the identity loss \mathcal{L}_{id} that computes the cosine similarity of the features extracted using the ArcFace [13]. Finally, to allow for refinement of the eye gaze direction, we calculate the gaze loss \mathcal{L}_g , which is the $L2$ distance between the gaze direction of the real image and the reconstructed image \hat{I} , extracted using a gaze estimation method [66]. To this end, our framework is able to refine the missing identity details from the initial reconstructed images.

Phase 2: Self reenactment During the second training phase, we train our model on self reenactment where the identity of the input source and target images is the same and the facial pose is different. Given the appearance of the source image I_s and the facial pose of the target image I_t , the hypernetwork is trained to predict the offsets $\Delta\theta$ so that the reenacted image I_r has the identity of the source image and the facial pose of the target image. The objective during this phase is:

$$\mathcal{L} = \lambda_{pix}\mathcal{L}_{pix} + \lambda_{lpips}\mathcal{L}_{lpips} + \lambda_{id}\mathcal{L}_{id} + \lambda_{sh}\mathcal{L}_{sh} + \lambda_g\mathcal{L}_g, \quad (3)$$

where \mathcal{L}_{pix} , \mathcal{L}_{lpips} , \mathcal{L}_{id} , and \mathcal{L}_g denote the losses described above, calculated between the target image I_t and the reenacted image I_r . In order to transfer the facial pose of the

target image, we calculate the shape loss \mathcal{L}_{sh} , i.e., the ℓ_1 distance of the 3D facial shapes extracted using [16] from the target and the reenacted image.

Phase 3: Cross-subject reenactment To further enhance our results on cross-subject reenactment, we propose to fine-tune our model trained on self reenactment using cross-subject image pairs. As cross-subject training is a challenging task, to ease the training process, we concurrently perform training for the tasks of self and cross-subject reenactment, by letting half of the image pairs in each batch to correspond to each task. Our training objective is the same as the one defined in (3), however, for the cross-subject batch samples we only calculate the identity \mathcal{L}_{id} loss between the source and the reenacted images and the shape loss \mathcal{L}_{sh} between the target and the reenacted images.

4. Experiments

In this section, we provide the implementation details and we present our quantitative and qualitative results and comparisons with state-of-the-art methods. In Section 4.1, we report results on self and cross-subject reenactment on the VoxCeleb1 [30] dataset, and in Section 4.2, we provide ablation studies to investigate the contribution of each design choice to the overall effectiveness of our method.

Implementation details We use the StyleGAN2 model [25] and the e4e inversion model [47] trained on the VoxCeleb1 dataset provided by [9] and we train our model on the same dataset with 256×256 images. We note that the only trainable modules of our method are the the Reenactment Module (RM) and the hypernetwork \mathcal{H} , while the encoders \mathcal{E}_w , \mathcal{E}_{app} , and \mathcal{E}_p , along with the StyleGAN2 generator \mathcal{G} are kept frozen (Fig. 2). We also note that we learn offsets for all the layers of the StyleGAN2 generator, except for the “toRGB” layers that mainly change the texture and the color of images [56] (please see the supplementary material for more details). We first train our model on the task of real image inversion with learning rate $2 \cdot 10^{-4}$ and we continue on the task of self reenactment with the same learning rate and a batch size of 16. We finally fine-tune our model on cross-subject reenactment with a constant learning rate of 10^{-4} . We set $\lambda_{pix} = 10.0$, $\lambda_{lpips} = 5.0$, $\lambda_{id} = 10.0$, $\lambda_{sh} = 0.5$ and $\lambda_g = 2.0$. All models are optimized with Adam optimizer [27] and are implemented in PyTorch [34].

4.1. Comparison with state-of-the-art methods

We evaluate our method on the test set of the VoxCeleb1 [30] dataset and we provide additional quantitative and qualitative results on the test set of the VoxCeleb2 [12] dataset in the supplementary material. We compare our framework with 10 state-of-the-art methods that have made their source code and models publicly available, namely, X2Face [55], FOMM [43], Neural [11], Fast BL [63],

Method	Self Reenactment						Cross-subject Reenactment			User Pref. (%)
	CSIM \uparrow	LPIPS \downarrow	FID \downarrow	FVD \downarrow	APD \downarrow	AED \downarrow	CSIM \uparrow	APD \downarrow	AED \downarrow	
X2Face [55]	<u>0.70</u>	0.21	<u>25.6</u>	490	1.3	9.0	0.57	2.2	16.4	-
FOMM [43]	0.64	0.27	35.3	523	4.6	12.6	0.53	10.9	20.9	-
Neural [11]	0.40	0.42	127.0	617	1.2	8.8	0.34	1.8	15.3	-
Fast BL [63]	0.65	0.41	55.0	706	<u>1.0</u>	7.6	0.58	1.4	14.7	5.9
PIR [37]	0.69	<u>0.23</u>	50.5	545	1.9	9.7	0.61	2.4	15.4	1.1
LSR [29]	0.59	0.26	63.0	484	<u>1.0</u>	7.5	0.50	1.5	13.1	5.0
FD [9]	0.65	<u>0.23</u>	19.0	400	<u>1.0</u>	6.5	0.49	1.7	10.2	4.1
LIA [54]	0.64	0.26	31.7	510	4.7	11.4	0.57	2.8	15.7	5.9
Dual [21]	0.26	0.39	46.5	600	3.4	12.5	0.19	3.1	16.9	-
Rome [26]	0.69	0.43	39.2	800	1.5	<u>5.6</u>	<u>0.63</u>	<u>1.2</u>	8.8	<u>10.2</u>
Ours	0.71	<u>0.23</u>	27.1	<u>480</u>	0.5	5.1	0.68	0.5	<u>9.3</u>	67.8

Table 1: Quantitative results on self and cross-subject reenactment. For CSIM metric, higher is better (\uparrow), while for the rest of the metrics lower is better (\downarrow). We note that the best and second best results are shown in bold and underline respectively.



Figure 3: Qualitative results and comparisons on self (first 3 rows) and cross-subject reenactment (last 3 rows) on Vox-Celeb1 [30]. Our method is able to faithfully preserve the identity characteristics and also effectively transfer the head pose, the facial expression and the gaze direction of the target face without producing significant visual artifacts.

PIR [37], LSR [29], FD [9], LIA [54], Dual [21], and Rome [26]. In the supplementary material, we provide additional comparisons with StyleHEAT [62] and StyleMask [10]. We also note that regarding the methods of Neural [11], LSR [29], FD [9] and Dual [21], we perform one-shot fine-tuning on a single source frame from each test video for fair comparisons against the proposed method.

Quantitative results We evaluate our method on two tasks,

namely, self reenactment and cross-subject reenactment. For self reenactment, we calculate six different evaluation metrics. Specifically, we measure the identity preservation by calculating the cosine similarity (CSIM) of the features extracted using the ArcFace face recognition network [13], and the reconstruction quality using the Learned Perceptual Image Path Similarity (LPIPS) metric [23]. Additionally, we calculate the Fréchet-Inception Distance (FID) [20]

to measure the quality of the reenacted images and the Fréchet-Video Distance (FVD) [50, 44] to measure the temporal consistency of the generated videos. Finally, to evaluate the facial pose transfer, we calculate the Average Pose Distance (APD) and the Average Expression Distance (AED), similarly to [37]. All metrics for self reenactment are calculated between the target and the reenacted images. When evaluating our method on the task of cross-subject reenactment, we calculate the CSIM metric between the source and the reenacted images, and we also calculate the APD and AED. In Table 1, we report results both on self and on cross-subject reenactment. We note that for self reenactment we report results on the test set of VoxCeleb1, where the first extracted frame from each video is used as the source frame and all other frames are used as the target ones. For cross-subject reenactment, we randomly select 35 video pairs from the test set of VoxCeleb1. In the task of self reenactment, our method outperforms all other techniques on identity preservation (CSIM), as well as on pose (APD) and facial expression (AED) transfer, while on LPIPS, FID and FVD metrics our method is among the best results. In the more challenging task of cross-subject reenactment, our method outperforms all other techniques on identity preservation and head pose transfer, while being on-par with Rome [26] on expression transfer.

Moreover, we conduct a user study to further assess the performance of the proposed method in comparison to state-of-the-art works. Specifically, we present 20 *randomly* selected image pairs, 10 of self and 10 of cross-subject reenactment, to 30 users and ask them to select the method that best reenacts the source frame in terms of (i) identity preservation, (ii) facial pose transfer, and (iii) image quality. We note that we opt to include only the methods that exhibit high performance on both quantitative and qualitative results. That is, we exclude X2Face [55] and FOMM [43], since they lead to several visual artifacts (as shown in the supplementary material). Similarly, we exclude Neural [11] and Dual [21] due to their poor quantitative results. As shown in Table 1, our method is by far the most preferable.

Finally, in order to evaluate the proposed method under the challenging (and far more useful in real-world applications) condition of large variations between the head pose of the source and target frames, we build a small benchmark set for the task of self reenactment, containing pairs of images with large head pose differences. Specifically, from each video of the VoxCeleb1 test set we select 5 image pairs with head pose distance (measured as the average of the absolute differences between the 3 Euler angles) larger than 15°. We report results in Table 2. Our method outperforms all other techniques on identity preservation and head pose transfer, while ranking second and performing on par with the method of Rome [26] on expression transfer.

Qualitative results In Fig. 3, we show qualitative com-

Method	CSIM \uparrow	APD \downarrow	AED \downarrow
X2Face [55]	0.45	3.1	12.1
FOMM [43]	0.44	3.2	12.7
Neural [11]	0.38	1.5	8.9
Fast BL [63]	0.44	1.5	9.0
PIR [37]	0.40	4.0	11.9
LSR [29]	0.48	1.4	8.5
FD [9]	0.34	2.7	10.4
LIA [54]	0.43	3.0	9.9
Dual [21]	0.23	4.9	12.6
Rome [26]	<u>0.53</u>	<u>1.1</u>	5.8
Ours	0.58	0.9	<u>6.2</u>

Table 2: Quantitative results on self reenactment using a set of images with large head pose differences between the source and target faces (subset of the VoxCeleb1 test set).

parisons on self and cross-subject reenactment. We note that for better visualization we report results only with the best performing methods, namely, Fast BL [63], PIR [37], LSR [29], FD [9], LIA [54] and Rome [26]. On the supplementary material we present additional comparisons with all methods. As shown, our method is able to produce mostly artifact free images, successfully preserve the source identity characteristics and faithfully transfer the target facial pose (i.e. head pose orientation, facial expression and gaze direction), even on the challenging task of cross-subject reenactment and on extreme head pose differences.

4.2. Ablation studies

In this section, we report the results of the ablation studies we performed to assess the contribution of: (a) the proposed curriculum learning (CL) scheme, (b) fine-tuning on the task of cross-subject reenactment, and (c) the gaze loss. Regarding (a), i.e. the contribution of the proposed curriculum learning scheme (Section 3.3), we provide results both on self and on cross-subject reenactment with our final model trained using the curriculum learning (CL) scheme and with a model trained directly on the task of self and cross-subject reenactment. As shown in Table 3, our model trained with CL achieves better results on head pose (APD), expression transfer (AED), and identity preservation on self reenactment. This is also shown in Fig. 4, where using CL leads to results less visual artifacts on the reenacted images.

For (b), we compare our method with and without training with cross-subject data. As shown in Table 4, when fine-tuning our method with cross-subject data, our quantitative results with respect to identity preservation (CSIM) are improved especially on cross-subject reenactment. The effect of cross-subject training on the identity preservation is also shown in Fig. 5, where we eliminate the identity leakage

Method	Self Reenactment			Cross Reenactment		
	CSIM	APD	AED	CSIM	APD	AED
w/o CL	0.69	0.7	6.7	0.72	0.6	11.6
w/ CL	0.71	0.5	5.1	0.68	0.5	9.3

Table 3: Quantitative results on self and cross-subject reenactment with and without using curriculum learning (CL).



Figure 4: Qualitative results on self and cross-subject reenactment with and without using the proposed CL scheme.



Figure 5: Qualitative results on cross-subject reenactment with and without training with cross-subject data (CSRT).

from the target images into the reenacted ones.

Method	CSIM \uparrow	
	SR	CR
w/o CSRT	0.69	0.53
w/ CSRT	0.71	0.68

Table 4: Quantitative results on self (SR) and cross-subject (CR) reenactment with and without performing cross-subject reenactment training (CSRT).

For (c), we perform experiments with and without using the gaze loss \mathcal{L}_g and we evaluate its contribution both quantitatively (Table 5) and qualitatively (Fig. 6). Specifically, in Table 5 we calculate the Gaze Error, i.e., the ℓ_2 distance between the gaze direction of the real and the generated images on three different tasks, namely, real image inversion (I), self reenactment (SR), and cross-subject reenactment

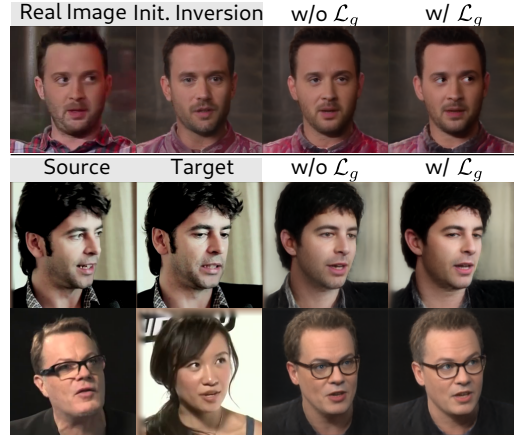


Figure 6: Illustration of the impact of the gaze loss \mathcal{L}_g on the generated images, on the tasks of real image inversion and self/cross-subject reenactment.

(CR). Moreover, in Fig. 6 we present examples of real image inversion (first row) and self and cross-subject reenactment (last two rows), with and without the gaze loss, where we observe that using the gaze loss improves the faithful reconstruction of the gaze direction of the target images.

Method	Gaze Error \downarrow		
	I	SR	CR
w/o \mathcal{L}_g	0.34	0.35	0.40
w/ \mathcal{L}_g	0.24	0.25	0.31

Table 5: Ablation on the impact of gaze loss \mathcal{L}_g . The gaze error is calculated on real image inversion (I), self reenactment (SR) and cross-subject reenactment (CR) tasks.

5. Conclusions

In this paper, we present HyperReenact, a method for neural face reenactment that jointly learns to refine and re-target the facial images using a pretrained StyleGAN2 model and a hypernetwork. We leverage the effectiveness of hypernetworks on the real image inversion task and extend their use for real image manipulation. Our method learns to fuse the disentangled representations of source identity and target facial pose, to effectively modify the weights of the generator, performing both identity refinement and facial pose re-targeting. We show that our approach can successfully reenact a source face, preserving the identity and transferring the target facial pose. We also demonstrate that our method can produce artifact-free images even on challenging cases of extreme head pose movements.

Acknowledgments: This work was supported by the EU H2020 AI4Media No. 951911 project.

A. Supplementary Material

In this supplementary material, we first provide a detailed analysis of the network architecture of the proposed framework (HyperReenact) in Section A.1. In Section A.2, we show results of two state-of-the-art inversion methods, namely, HyperStyle [4] and HyperInverter [14], on real image editing and compare our method against HyperStyle [4]. In Section A.3, we discuss the limitations of our method and in Section A.4 we present comparisons against two methods, namely, StyleHEAT [62] and StyleMask [10], while in Section A.6 we provide additional results of our method on the VoxCeleb1 [30] and the VoxCeleb2 [12] datasets. Finally, along with this report, please find attached an external video file that includes 10 randomly selected identities for self reenactment and 10 randomly selected pairs for cross-subject reenactment (for both VoxCeleb1 [30] and VoxCeleb2 [12] datasets).

A.1. Network architecture

In this section we provide details of the various components of the proposed framework (HyperReenact). An overview of HyperReenact is shown in Fig. 2 in the main paper. More specifically, we propose to blend the appearance feature map f_{app} of size $512 \times 7 \times 7$ and the pose feature map f_p of size $2048 \times 7 \times 7$ using the Reenactment Module (RM), which is illustrated in Fig. 7. As shown, we first project the f_p using a convolutional layer (kernel=(1,1), stride=1, pad=0) into the same channel size of f_{app} . Then for each feature map we learn the two modulation parameters γ and β using convolutions with a kernel size of 1, stride set to 1 and padding set to 0. The output feature map f_r of size $512 \times 7 \times 7$, computed using Eq. 1 from the main paper, is then fed into the different reenactment blocks of the hypernetwork.

Our hypernetwork \mathcal{H} consists of $M < N$ Reenactment Blocks (RB), where M is the number of generator layers that we control and N is the total number of layers in the generator. Each of those blocks takes as input the feature map f_r and outputs the corresponding offset $\Delta\theta_\ell$ for the weights of each layer of the generator. In Table 6 we report the structure of StyleGAN2 generator trained on 256×256 image resolution. From the blocks shown in Table 6 we only modify the convolutional layers (Conv), as the ToRGB layers mainly affect the colors of the generated images [56]. Hence, we propose to modify $M = 13$ layers.

In order to reduce the number of trainable parameters, similarly to HyperStyle [4], we propose to use two types of Reenactment Blocks, namely, the Shared Reenactment Blocks and the Layer-specific Reenactment Blocks, as shown in Fig. 7. We present an overview of the architecture of both blocks in Fig. 8. In both types of blocks, the input feature map f_r is first fed into a series of convolu-

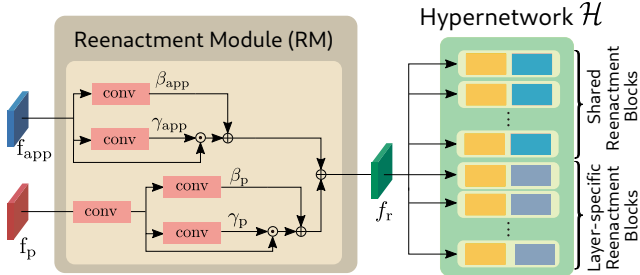


Figure 7: Illustration of the two learnable components of our architecture, namely, the Reenactment Module (RM) and the hypernetwork \mathcal{H} . The RM module fuses appearance features from the source face and outputs the fused feature map f_r , that drives the hypernetwork \mathcal{H} . The hypernetwork \mathcal{H} consists of multiple Reenactment Blocks with each one of them corresponding to a particular layer of the generator.

tional layers that process and down-sample the input into the shape of $512 \times 1 \times 1$. Regarding the Shared Reenactment Blocks, as shown on the top row of Fig. 8, the down-sampled feature map is then flattened and fed into a fully-connected layer. Afterwards, the two shared fully-connected layers are used to calculate the output feature map of shape $C_\ell^{out} \times C_\ell^{in} \times 1 \times 1$, which is repeated spatially so as to match the shape of the convolutional kernels ($C_\ell^{out} \times C_\ell^{in} \times k_\ell \times k_\ell$). Regarding the Layer-specific Reenactment Blocks, as shown on the bottom row of Fig. 8, after the series of down-sampling convolutional layers, the computed feature map has a shape of $512 \times 1 \times 1$. This feature map, upon being flattened, is fed into a final fully-connected layer which outputs a feature map of shape $C_\ell^{out} \times C_\ell^{in} \times 1 \times 1$, which is then spatially repeated to have a shape of $C_\ell^{out} \times C_\ell^{in} \times k_\ell \times k_\ell$.

We provide in more detail the structure of the two Reenactment Blocks in Tables 7 and 8. Finally, in Table 9, we report the StyleGAN2 layers that we propose to modify as well as the type of the Reenactment Block that we use for each layer. As shown in Table 9, we use the Shared Reenactment Blocks for the first seven layers of the generator. As a result, the parameters of two fully-connected layers across all the Shared Reenactment Blocks, are common. For the last six layers of the generator, we use the Layer-specific Reenactment Blocks. We note that the use of the shared layers reduces our trainable parameters from 1.2B to 300M.

A.2. Comparisons with HyperStyle

As discussed in the main paper, the proposed framework draws inspiration from two state-of-the-art methods (for the task of real image inversion), namely, HyperStyle [4] and HyperInverter [14]. Specifically, similarly to [4, 14],

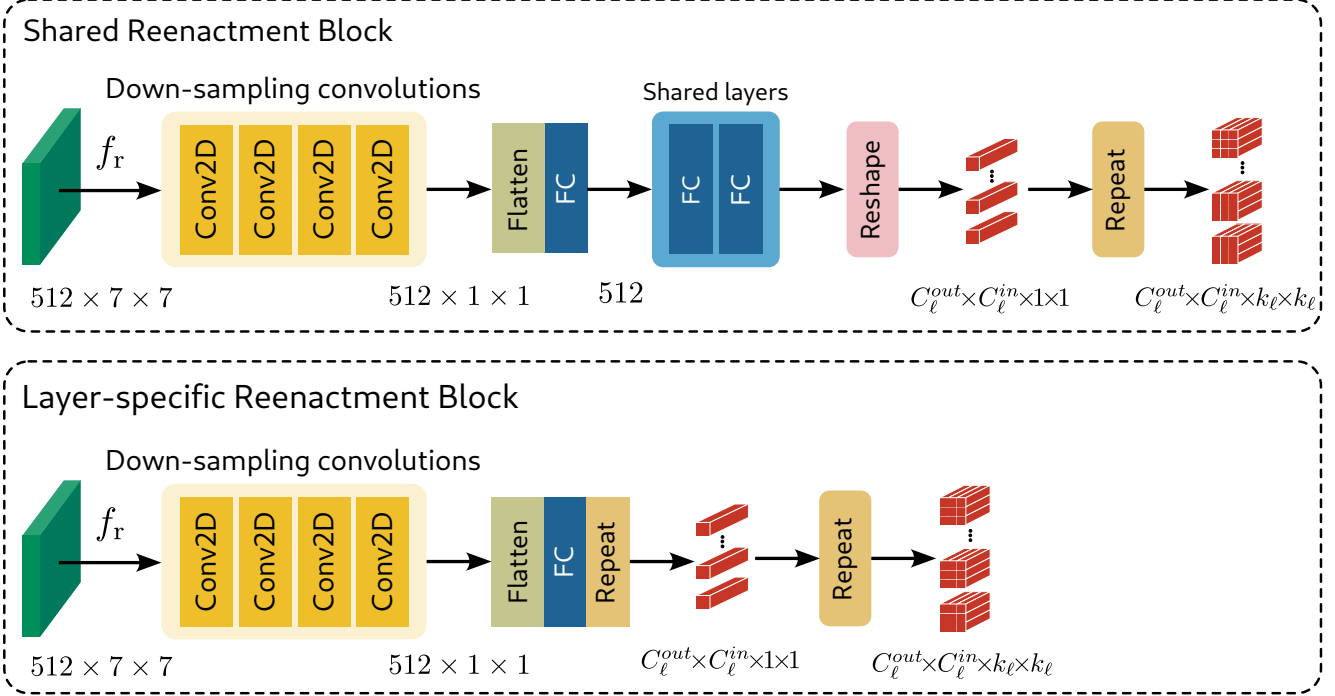


Figure 8: Detailed architecture of the two types of Reenactment Blocks, namely the Shared Reenactment Block (top row) and the Layer-specific Reenactment Block (bottom row).

Layer ℓ Index	Layer ℓ Name	Resolution	Kernel Dim. $C_\ell^{out} \times C_\ell^{in} \times k_\ell \times k_\ell$
0	Conv1	4×4	$512 \times 512 \times 3 \times 3$
1	ToRGB1	4×4	$3 \times 512 \times 1 \times 1$
2	Conv2	8×8	$512 \times 512 \times 3 \times 3$
3	Conv3	8×8	$512 \times 512 \times 3 \times 3$
4	ToRGB2	8×8	$3 \times 512 \times 1 \times 1$
5	Conv4	16×16	$512 \times 512 \times 3 \times 3$
6	Conv5	16×16	$512 \times 512 \times 3 \times 3$
7	ToRGB3	16×16	$3 \times 512 \times 1 \times 1$
8	Conv6	32×32	$512 \times 512 \times 3 \times 3$
9	Conv7	32×32	$512 \times 512 \times 3 \times 3$
10	ToRGB4	32×32	$3 \times 512 \times 1 \times 1$
11	Conv8	64×64	$256 \times 512 \times 3 \times 3$
12	Conv9	64×64	$256 \times 256 \times 3 \times 3$
13	ToRGB5	64×64	$3 \times 256 \times 1 \times 1$
14	Conv10	128×128	$128 \times 256 \times 3 \times 3$
15	Conv11	128×128	$128 \times 128 \times 3 \times 3$
16	ToRGB6	128×128	$3 \times 128 \times 1 \times 1$
17	Conv12	256×256	$64 \times 128 \times 3 \times 3$
18	Conv13	256×256	$64 \times 64 \times 3 \times 3$
19	ToRGB7	256×256	$3 \times 64 \times 1 \times 1$

Table 6: Structure of StyleGAN2 generator trained on 256×256 image resolution.

Layer	Input shape	Output shape
Conv2D kernel=(3,3), stride=1, pad=1	$B \times 512 \times 7 \times 7$	$B \times 128 \times 7 \times 7$
LeakyReLU activation slope=0.01	$B \times 128 \times 7 \times 7$	$B \times 128 \times 7 \times 7$
Conv2D kernel=(3,3), stride=1, pad=0	$B \times 128 \times 7 \times 7$	$B \times 128 \times 5 \times 5$
LeakyReLU activation slope=0.01	$B \times 128 \times 5 \times 5$	$B \times 128 \times 5 \times 5$
Conv2D kernel=(3,3), stride=1, pad=0	$B \times 128 \times 5 \times 5$	$B \times 128 \times 3 \times 3$
LeakyReLU activation slope=0.01	$B \times 128 \times 3 \times 3$	$B \times 128 \times 3 \times 3$
Conv2D kernel=(3,3), stride=1, pad=0	$B \times 128 \times 3 \times 3$	$B \times 512 \times 1 \times 1$
LeakyReLU activation slope=0.01	$B \times 512 \times 1 \times 1$	$B \times 512 \times 1 \times 1$
Flatten	$B \times 512 \times 1 \times 1$	$B \times 512$
FC	$B \times 512$	$B \times 512$
Shared FC	$B \times 512$	$B \times (512 \times 512)$
Shared FC	$B \times (512 \times 512)$	$B \times (512 \times 512)$
Reshape	$B \times (512 \times 512)$	$B \times 512 \times 512 \times 1 \times 1$
Repeat	$B \times 512 \times 512 \times 1 \times 1$	$B \times 512 \times 512 \times 3 \times 3$

Table 7: Architecture of the Shared Reenactment Blocks (B denotes the batch size).

Layer	Input shape	Output shape
Conv2D kernel=(3,3), stride=1, pad=1	$B \times 512 \times 7 \times 7$	$B \times 256 \times 7 \times 7$
LeakyReLU activation slope=0.01	$B \times 256 \times 7 \times 7$	$B \times 256 \times 7 \times 7$
Conv2D kernel=(3,3), stride=1, pad=0	$B \times 256 \times 7 \times 7$	$B \times 256 \times 5 \times 5$
LeakyReLU activation slope=0.01	$B \times 256 \times 5 \times 5$	$B \times 256 \times 5 \times 5$
Conv2D kernel=(3,3), stride=1, pad=0	$B \times 256 \times 5 \times 5$	$B \times 256 \times 3 \times 3$
LeakyReLU activation slope=0.01	$B \times 256 \times 3 \times 3$	$B \times 256 \times 3 \times 3$
Conv2D kernel=(3,3), stride=1, pad=0	$B \times 256 \times 3 \times 3$	$B \times 512 \times 1 \times 1$
LeakyReLU activation slope=0.01	$B \times 512 \times 1 \times 1$	$B \times 512 \times 1 \times 1$
Flatten	$B \times 512 \times 1 \times 1$	$B \times 512$
FC	$B \times 512$	$B \times (C_\ell^{out} \times C_\ell^{in})$
Reshape	$B \times (C_\ell^{out} \times C_\ell^{in})$	$B \times C_\ell^{out} \times C_\ell^{in} \times 1 \times 1$
Repeat	$B \times C_\ell^{out} \times C_\ell^{in} \times 1 \times 1$	$B \times C_\ell^{out} \times C_\ell^{in} \times 3 \times 3$

Table 8: Architecture of the Layer-specific Reenactment Blocks. The input on the block has a size of $B \times 512 \times 7 \times 7$, where B is the batch size, while C_ℓ^{in} and C_ℓ^{out} are the input and output channels, respectively.

Layer Index ℓ	ℓ -th Layer Name	RB Type
0	Conv1	Shared
2	Conv2	Shared
3	Conv3	Shared
5	Conv4	Shared
6	Conv5	Shared
8	Conv6	Shared
9	Conv7	Shared
11	Conv8	Layer-specific
12	Conv9	Layer-specific
14	Conv10	Layer-specific
15	Conv11	Layer-specific
17	Conv12	Layer-specific
18	Conv13	Layer-specific

Table 9: Convolutional layers of the StyleGAN2 generator that we propose to modify, altering their weights using the offsets computed by the hypernetwork.

we also incorporate a hypernetwork [18] in order to learn how to effectively modify the weights of a pretrained StyleGAN2 [25] generator. However, we note that [4, 14] aim at the problem of real image inversion, not neural face reenactment.

Both HyperStyle [4] and HyperInverter [14] produce high-quality results on real image inversion, however their quality degenerates significantly when manipulating the inverted images, especially on head pose editing. In Fig. 9, we show results of HyperStyle [4] (Fig. 9a) and HyperInverter [14] (Fig. 9b) on head pose editing using the CelebA dataset [24]. It is worth noting that, while both methods excel on real image inversion (the inverted images are inside the red boxes), they produce several visual artifacts on the edited images, which renders them unsuitable for the task of real image reenactment. We note that we obtain the edited images using the InterFaceGAN method [42] to shift the latent codes.

To further compare our method with HyperStyle [4], instead of simply editing the head pose as shown above, we perform face reenactment using the learned directions from the work of FD [9]. We note that FD learns the directions in the W^+ latent space of a StyleGAN2 model trained on the VoxCeleb1 [30] dataset that are responsible for controlling

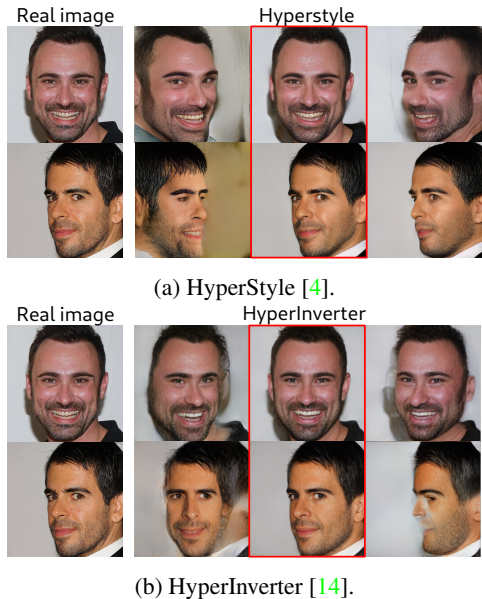


Figure 9: Real image inversion and editing results on CelebA dataset [24] using HyperStyle [4] and HyperInverter [14]. Inside the red boxes are the inverted images, while on the right and left we show results of head pose editing using the method of InterFaceGAN [42].

the facial pose. In order to test HyperStyle along with FD, we train HyperStyle on the VoxCeleb1 dataset. We refer to this pipeline as HyperStyle-FD. In Fig. 10 we show comparisons of our method against HyperStyle-FD. We note that the reenacted images using HyperStyle-FD not only present visual artifacts, but also look unnatural, especially when the source and target images have large head pose differences.

A.3. Limitations

As shown in the main paper and in the additional experimental results provided in this supplementary material, the proposed HyperReenact, in contrast to several state-of-the-art works, achieves to effectively reenact a source face given a target facial pose, preserving the source identity characteristics and producing artifact-free images, especially in the cases where the target and the source faces differ largely in head pose. Nevertheless, we observe that in cases where the source facial images depict accessories such as hats or eyeglasses, the proposed method fails to reconstruct them in detail. For instance, as shown in Fig. 11, our method cannot fully reconstruct the style of the glasses in the example of the first row. Similarly, regarding the examples of the second and third row of Fig. 11, our method is not able to reconstruct every detail on the hats. We attribute this to the fact that such items are underrepresented on the VoxCeleb1 dataset and, as a result, our method is not able to learn how

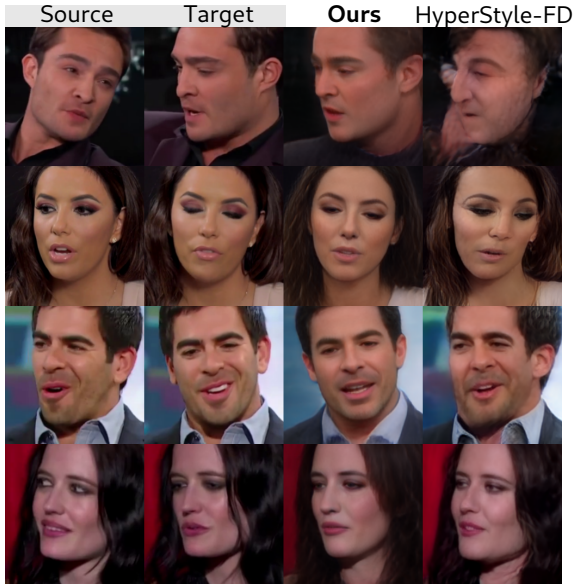


Figure 10: Qualitative comparisons against HyperStyle-FD.



Figure 11: We observe that accessory items such as hats or glasses are not fully reconstructed.

to reconstruct them. Additionally, we do not refine any details on the background of the generated images.

A.4. Comparisons with StyleHEAT [62] and StyleMask [10]



Figure 12: Qualitative comparisons with StyleHEAT [62] and StyleMask [10] on VoxCeleb dataset [30].

As discussed in Section 4, StyleHEAT [62] is trained on the HDTF dataset [67], that consists of facial images exhibiting only small roll angle variations and showing mostly frontal views. Moreover, StyleMask [10] is a face reenactment method based on a pretrained StyleGAN2 model trained on the FFHQ dataset, which learns to disentangle the identity characteristics from the facial pose using the disentangled properties of the style space \mathcal{S} of StyleGAN2. Both StyleHEAT and StyleMask require the input images to be aligned, similarly to the FFHQ dataset [24]. In Fig. 12 and in Table 10, we present qualitative and quantitative comparisons with StyleHEAT and StyleMask on the VoxCeleb1 dataset [30]. Clearly, StyleHEAT performs poorly by generating many visual artifacts when the source and target images have large pose variations, while StyleMask is not able to faithfully reconstruct the source identity characteristics. For a fair comparison, we additionally compare on the HDTF dataset [67] (where StyleHEAT has been trained on). In Table 11, we provide quantitative results on the test videos provided by the authors of StyleHEAT [62]. Finally, Fig. 13 illustrates qualitative comparisons with both StyleHEAT and StyleMask. As shown, our method evidently outperforms both StyleHEAT and StyleMask, on identity preservation and on facial pose transfer.

A.5. Benchmark with extreme head pose differences

We build a small benchmark in order to evaluate our method on challenging cases where the source and target faces have large head pose differences. Specifically, considering the VoxCeleb1 [30] test dataset, we pick 1000 pairs of images with large head pose distance (measured as the average of the absolute differences between the 3 Euler an-

Method	CSIM \uparrow	APD \downarrow	AED \downarrow
StyleHEAT [62]	0.45	8.6	12.9
StyleMask [10]	0.47	5.3	13.2
Ours	0.58	0.9	6.2

Table 10: Quantitative comparisons with StyleHEAT [62] and StyleMask [10] on the small benchmark with large head pose differences between the source and target faces.

Method	CSIM \uparrow	APD \downarrow	AED \downarrow
StyleHEAT [62]	0.72	1.1	7.5
StyleMask [10]	0.66	1.6	8.8
Ours	0.75	0.38	4.1

Table 11: Quantitative comparisons on self-reenactment with StyleHEAT [62] and StyleMask [10] on HDTF dataset [67]



Figure 13: Qualitative comparisons with StyleHEAT [62] and StyleMask [10] on HDTF dataset [67].

gles). In Fig. 15 we show the distribution of the absolute pose differences for each Euler angle, namely yaw, pitch and roll. This benchmark allows us to obtain deeper insights on the behavior of reenactment methods on challenging conditions.

In Fig. 14, we show comparisons of our method against the two state-of-the-art face reenactment methods, namely Fast BL [63] and Rome [26], on image pairs selected from the small benchmark described above. As shown, the source and target images have extreme head pose differences which makes it more challenging for face reenactment methods to generate realistic images. Nevertheless, our method is able



Figure 14: Qualitative results on the small benchmark with large head pose differences between the source and target faces.

to synthesize realistic faces even on extreme head poses. In Fig. 14 we highlight (red boxes) details on the human faces, such as areas around the mouth and eyes, where our method creates artifact free results, while Fast BL [63] and Rome [26] generate blurry unrealistic images.

A.6. Additional quantitative/qualitative results

In this Section, we present additional quantitative and qualitative results of the proposed method in comparison to state-of-the-art works. We compare all methods in terms of their inference time and their overall performance on the two tasks, namely, self and cross-subject reenactment. Specifically, in Table 12, we demonstrate the inference time of each method while reenacting a video of 200 frames. To help drawing connections between the inference time and the performance of each method, we also report the performance ranking in Table 12 (referred to as ‘‘Perf. Rank’’). Specifically, we consider the evaluation metrics that are reported in Table 1 of the main paper, and rank all methods with respect to each metric. Then, we average the ranking positions across all metrics on both self and cross-subject reenactment, for each method, to obtain its overall performance ranking. In Fig. 16, we present a plot of the two metrics (Inference time and Overall Performance Ranking). We note that our method achieves the best overall performance ranking, while also remaining competitive on the

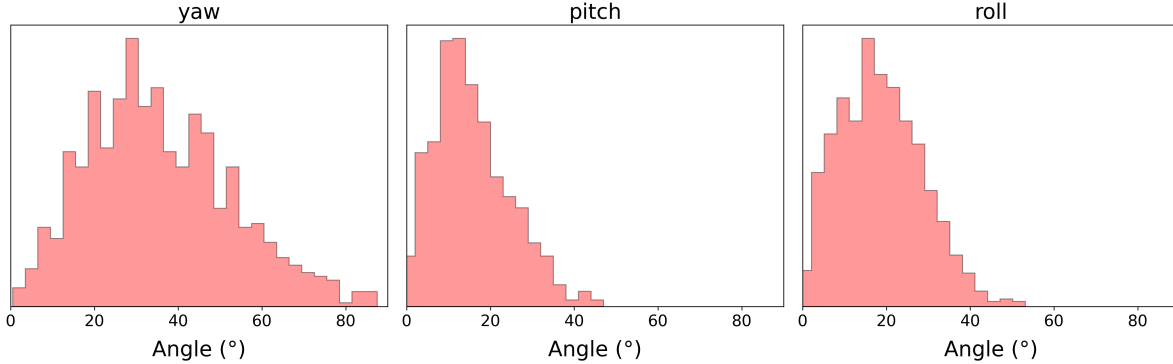


Figure 15: Distribution of the absolute pose differences for each Euler angle (yaw, pitch and roll) in our small benchmark dataset built from VoxCeleb1 test dataset.

inference time. Additionally, while the inference time of X2Face [55] and FOMM [43] is low compared to the other methods, as shown from the qualitative results they generate images with several visual artifacts.

Moreover, we present further qualitative comparisons on the VoxCeleb1 [30], as well as quantitative and qualitative results on VoxCeleb2 [12]. Specifically, in Figs. 17 and 18 we show results on self reenactment, in Figs. 19 and 20 we show results on cross-subject reenactment, and in Fig. 21 we report results on self reenactment on the benchmark with extreme head pose differences described in Sect. A.5.

Additionally, we quantitatively compare our method on the task of self reenactment on the VoxCeleb2 dataset [12] with the 10 state-of-the-art methods, namely, X2Face [55], FOMM [43], Neural [11], Fast BL [63], PIR [37], LSR [29], FD [9], LIA [54], Dual [21], and Rome [26]. In Table. 13, we present the quantitative results on self reenactment. As shown, our method outperforms all other methods both on identity preservation (CSIM) and on head pose (APD) and expression (AED) transfer (similarly to Section 4.1).

Finally, in Figs. 22, 23, we demonstrate results of our method, both on self and on cross-subject reenactment, on additional video datasets, namely, FaceForensics [40], 300-VW [41], and CelebV-HQ [68], showing that the proposed method can generalise well on different video benchmarks.

A.7. Ethics consideration

Neural face reenactment methods allow for the creation of realistic talking head sequences that resemble the real ones. Consequently, besides being used for benevolent and useful purposes, such as in video conferencing, film and video production, arts, and education, we acknowledge that face reenactment methods, such as the proposed one, can also be misused towards malevolent purposes, such as deep-fake fraud, that can harm individuals and can pose a greater societal threat.

Method	Inf. time (sec)↓	Perf. Rank ↓
X2Face [55]	11.0	3.5
FOMM [43]	11.0	6.3
Neural [11]	115.0	5.6
Fast BL [63]	61.0	4.0
PIR [37]	54.0	4.3
LSR [29]	110.0	4.2
FD [9]	40.0	<u>2.7</u>
LIA [54]	<u>23.0</u>	5.3
Dual [21]	<u>23.0</u>	6.9
Rome [26]	70.0	3.1
Ours	37.0	1.1

Table 12: Quantitative comparisons on inference time and overall performance ranking (Perf. Rank) of all methods on self and cross-subject reenactment tasks (inference time is calculated while reenacting a video of 200 frames).

Method	CSIM↑	APD↓	AED↓
X2Face [55]	0.60	2.4	10.6
FOMM [43]	0.57	5.1	13.6
Neural [11]	0.39	1.4	9.1
Fast BL [63]	0.57	1.1	8.6
PIR [37]	0.57	2.8	10.8
LSR [29]	0.61	<u>1.0</u>	7.5
FD [9]	0.59	1.3	7.3
LIA [54]	0.64	2.5	8.7
Dual [21]	0.15	3.7	12.7
Rome [26]	<u>0.63</u>	1.3	<u>5.9</u>
Ours	0.65	0.5	5.2

Table 13: Quantitative results on the task of self reenactment on VoxCeleb2 dataset [12].

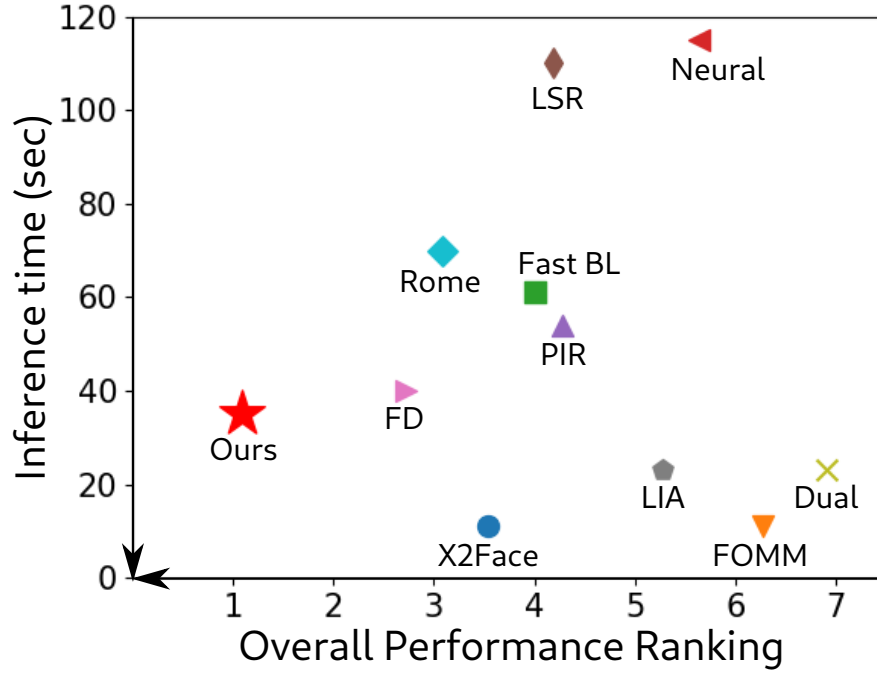


Figure 16: Comparisons in terms of the overall performance ranking of each method (presented in Table 1 of the main paper) and their inference time required to reenact a video of 200 frames. The arrows point towards the best results.

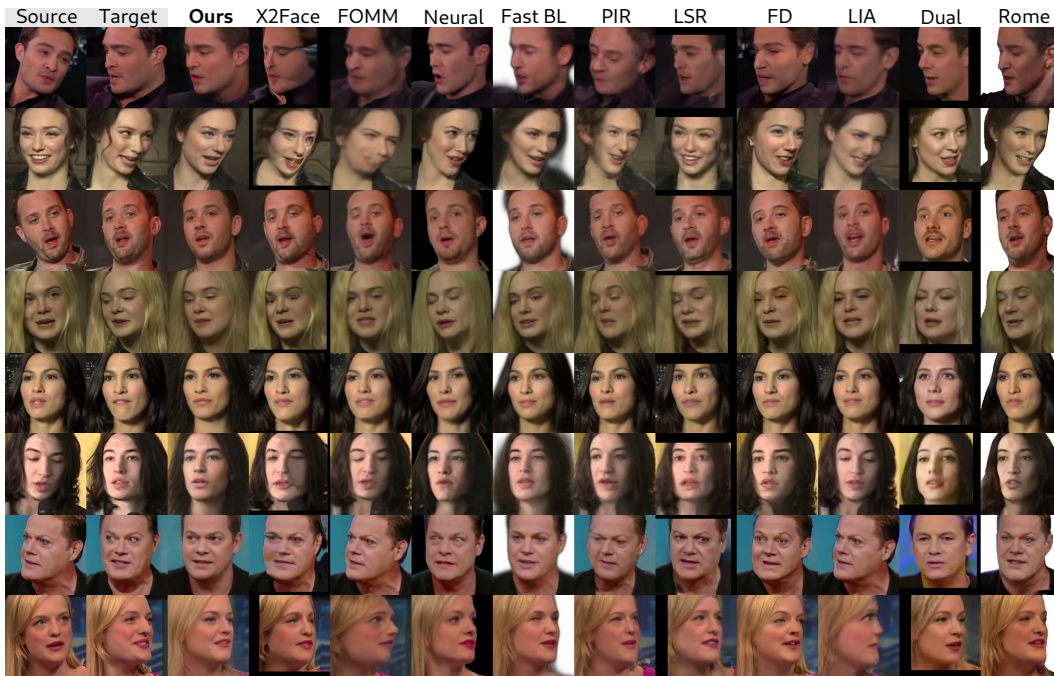


Figure 17: Additional qualitative results and comparisons on self-reenactment on VoxCeleb1 dataset [30]. The first and second columns show the source and target faces.



Figure 18: Qualitative results and comparisons on self-reenactment on VoxCeleb2 dataset [12]. The first and second columns show the source and target faces.



Figure 19: Additional qualitative results and comparisons on cross-subject reenactment on VoxCeleb1 dataset [30]. The first and second columns show the source and target faces.

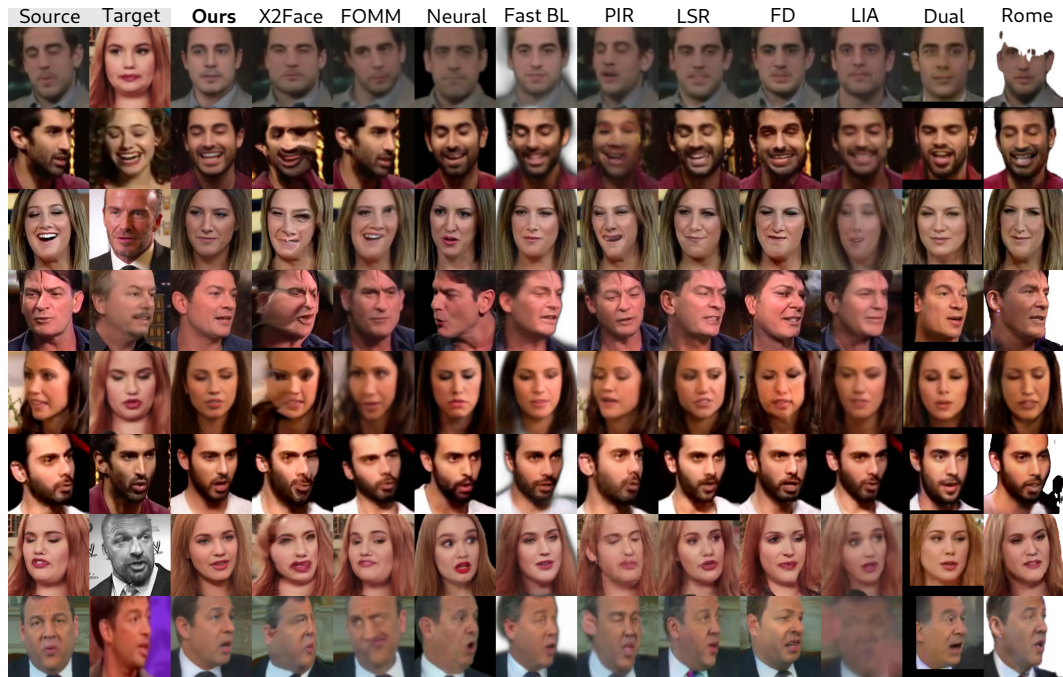


Figure 20: Qualitative results and comparisons on cross-subject reenactment on VoxCeleb2 dataset [12]. The first and second columns show the source and target faces, which are from different identities.



Figure 21: Additional qualitative results and comparisons on self reenactment using our small benchmark with image pairs that have large head pose differences. We show that our method presents robust results with artifact-free images, compared to the other state-of-the-art methods.



Figure 22: Additional qualitative results of our method on other video datasets such as FaceForensics [40] and 300-VW [41].



Figure 23: Additional qualitative results of our method on CelebV-HQ video dataset [68].

References

- [1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. [3](#)
- [2] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (ToG)*, 40(3), 2021. [2](#)
- [3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6711–6720, 2021. [3](#)
- [4] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18511–18521, 2022. [2](#), [3](#), [4](#), [9](#), [12](#)
- [5] Qingyan Bai, Yinghao Xu, Jiapeng Zhu, Weihao Xia, Yujun Yang, and Yujun Shen. High-fidelity gan inversion with padding space. In *European Conference on Computer Vision*, pages 36–53. Springer, 2022. [3](#)
- [6] Simone Barattin, Christos Tzelepis, Ioannis Patras, and Nicu Sebe. Attribute-preserving face dataset anonymization via latent code optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8001–8010, 2023. [2](#)
- [7] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. [5](#)
- [8] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999. [3](#)
- [9] Stella Bounareli, Vasileios Argyriou, and Georgios Tzimiropoulos. Finding directions in gan’s latent space for neural face reenactment. *British Machine Vision Conference (BMVC)*, 2022. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [12](#), [15](#)
- [10] Stella Bounareli, Christos Tzelepis, Vasileios Argyriou, Ioannis Patras, and Georgios Tzimiropoulos. Stylemask: Disentangling the style space of stylegan2 for neural face reenactment. *IEEE Conference on Automatic Face and Gesture Recognition*, 2023. [2](#), [3](#), [6](#), [9](#), [13](#), [14](#)
- [11] Egor Burkov, Igor Pasechnik, Artur Grigorev, and Victor Lempitsky. Neural head reenactment with latent pose descriptors. In *CVPR*, 2020. [2](#), [5](#), [6](#), [7](#), [15](#)
- [12] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018. [2](#), [3](#), [5](#), [9](#), [15](#), [17](#), [18](#)
- [13] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. [4](#), [5](#), [6](#)
- [14] Tan M Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11389–11398, 2022. [2](#), [3](#), [9](#), [12](#)
- [15] Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. Headgan: One-shot neural head synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14398–14407, 2021. [3](#)
- [16] Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021. [3](#), [4](#), [5](#)
- [17] Partha Ghosh, Pravir Singh Gupta, Roy Uziel, Anurag Ranjan, Michael J Black, and Timo Bolkart. Gif: Generative interpretable faces. In *2020 International Conference on 3D Vision (3DV)*, pages 868–878. IEEE, 2020. [3](#)
- [18] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*, 2017. [2](#), [3](#), [12](#)
- [19] Sungjoo Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. Marionette: Few-shot face reenactment preserving identity of unseen targets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 10893–10900, 2020. [3](#)
- [20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. [6](#)
- [21] Gee-Sern Hsu, Chun-Hung Tsai, and Hung-Yi Wu. Dual-generator face reenactment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 642–650, 2022. [2](#), [6](#), [7](#), [15](#)
- [22] Po-Hsiang Huang, Fu-En Yang, and Yu-Chiang Frank Wang. Learning identity-invariant motion representations for cross-id face reenactment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7084–7092, 2020. [3](#)
- [23] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. [5](#), [6](#)
- [24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. [1](#), [3](#), [12](#), [13](#)
- [25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. [1](#), [2](#), [3](#), [5](#), [12](#)
- [26] Taras Khakhulin, Vanessa Sklyarova, Victor Lempitsky, and Egor Zakharov. Realistic one-shot mesh-based head avatars. In *European Conference on Computer Vision*, pages 345–362. Springer, 2022. [1](#), [2](#), [3](#), [6](#), [7](#), [14](#), [15](#)
- [27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun,

- editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [5](#)
- [28] Mohammad Rami Koujan, Michail Christos Doukas, Anastasios Roussos, and Stefanos Zafeiriou. Head2head: Video-based neural head synthesis. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 16–23. IEEE, 2020. [3](#)
- [29] Moustafa Meshry, Saksham Suri, Larry S Davis, and Abhinav Shrivastava. Learned spatial representations for few-shot talking-head synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13829–13838, 2021. [1](#), [2](#), [6](#), [7](#), [15](#)
- [30] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTER-SPEECH*, 2017. [2](#), [3](#), [4](#), [5](#), [6](#), [9](#), [12](#), [13](#), [15](#), [16](#), [17](#)
- [31] James Oldfield, Markos Georgopoulos, Yannis Panagakis, Mihalis A Nicolaou, and Ioannis Patras. Tensor component analysis for interpreting the latent space of gans. *British Machine Vision Conference (BMVC)*, 2021. [2](#)
- [32] James Oldfield, Christos Tzelepis, Yannis Panagakis, Mihalis A Nicolaou, and Ioannis Patras. Panda: Unsupervised learning of parts and appearances in the feature maps of gans. *International Conference on Learning Representations (ICLR)*, 2023. [2](#)
- [33] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019. [4](#)
- [34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. [5](#)
- [35] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. [2](#)
- [36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. [3](#)
- [37] Yurui Ren, Ge Li, Yuanqi Chen, Thomas H Li, and Shan Liu. Pirenderer: Controllable portrait image generation via semantic neural rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13759–13768, 2021. [1](#), [3](#), [6](#), [7](#), [15](#)
- [38] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021. [3](#)
- [39] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744*, 2021. [3](#)
- [40] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics: A large-scale video dataset for forgery detection in human faces. *arXiv*, 2018. [3](#), [15](#), [19](#)
- [41] Jie Shen, Stefanos Zafeiriou, Grigoris G Chrysos, Jean Kossaiji, Georgios Tzimiropoulos, and Maja Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 50–58, 2015. [15](#), [19](#)
- [42] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, 2020. [2](#), [12](#)
- [43] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in Neural Information Processing Systems*, 32:7137–7147, 2019. [3](#), [5](#), [6](#), [7](#), [15](#)
- [44] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3626–3636, 2022. [7](#)
- [45] Adéla Šubrťová, David Futschik, Jan Čech, Michal Lukáč, Eli Shechtman, and Daniel Šykora. Chunkygan: Real image inversion via segments. In *European Conference on Computer Vision*, pages 189–204. Springer, 2022. [3](#)
- [46] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6142–6151, 2020. [2](#), [3](#)
- [47] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021. [2](#), [3](#), [4](#), [5](#)
- [48] Christos Tzelepis, James Oldfield, Georgios Tzimiropoulos, and Ioannis Patras. Contraclip: Interpretable gan generation driven by pairs of contrasting sentences. *arXiv preprint arXiv:2206.02104*, 2022. [2](#)
- [49] Christos Tzelepis, Georgios Tzimiropoulos, and Ioannis Patras. Warpedganspace: Finding non-linear rbf paths in gan latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6393–6402, 2021. [2](#)
- [50] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. [7](#)
- [51] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*. PMLR, 2020. [2](#)
- [52] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute

- editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11379–11388, 2022. 3
- [53] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10039–10049, 2021. 3
- [54] Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. Latent image animator: Learning to animate images via latent space navigation. In *International Conference on Learning Representations*, 2021. 1, 3, 6, 7, 15
- [55] Olivia Wiles, A Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–686, 2018. 3, 5, 6, 7, 15
- [56] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 5, 9
- [57] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3
- [58] Chao Xu, Jiangning Zhang, Yue Han, Guanzhong Tian, Xianfang Zeng, Ying Tai, Yabiao Wang, Chengjie Wang, and Yong Liu. Designing one unified framework for high-fidelity face reenactment and swapping. In *European Conference on Computer Vision*, pages 54–71. Springer, 2022. 3
- [59] Huiting Yang, Liangyu Chai, Qiang Wen, Shuang Zhao, Zixun Sun, and Shengfeng He. Discovering interpretable latent space directions of gans beyond binary attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12177–12185, 2021. 2
- [60] Kewei Yang, Kang Chen, Daoliang Guo, Song-Hai Zhang, Yuanchen Guo, and Weidong Zhang. Face2face^P: Real-time high-resolution one-shot face reenactment. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XIII*. Springer, 2022. 3
- [61] Qi Yang, Christos Tzelepis, Sergey Nikolenko, Ioannis Patras, and Aleksandr Farseev. ”just to see you smile”: Smiley, a voice-guided guy gan. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 1196–1199, 2023. 2
- [62] Fei Yin, Yong Zhang, Xiaodong Cun, Mingdeng Cao, Yanbo Fan, Xuan Wang, Qingyan Bai, Baoyuan Wu, Jue Wang, and Yujiu Yang. Styleheat: One-shot high-resolution editable talking face generation via pretrained stylegan. *arXiv preprint arXiv:2203.04036*, 2022. 2, 3, 6, 9, 13, 14
- [63] Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. Fast bi-layer neural synthesis of one-shot realistic head avatars. In *ECCV*, 2020. 1, 2, 3, 5, 6, 7, 14, 15
- [64] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9459–9468, 2019. 2, 3
- [65] Jiangning Zhang, Xianfang Zeng, Mengmeng Wang, Yusu Pan, Liang Liu, Yong Liu, Yu Ding, and Changjie Fan. Freenet: Multi-identity face reenactment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5326–5335, 2020. 3
- [66] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *European Conference on Computer Vision*, pages 365–381. Springer, 2020. 5
- [67] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3661–3670, 2021. 3, 13, 14
- [68] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy. CelebV-HQ: A large-scale video facial attributes dataset. In *ECCV*, 2022. 15, 20